The Understanding and Generation of Ellipses in a Natural Language System.*

Lisa F. Rau

Berkeley Artificial Intelligence Research Project
Computer Science Division
University of California, Berkeley
Berkeley, California, 94720.

ABSTRACT

Ellipsis in English can occur in many different forms. A large class of ellipses is isolated, encompassing such traditionally distinct linguistic rules as Right-node Raising, Gapping, VP-Deletion, Coordinate Reduction, and other phenomena which crosses sentence boundaries. The members of this class of ellipses involve implicit or explicit coordination. A uniform algorithm implemented in the UC (Unix Consultant) natural language system resolves the ellipses in this class, and constraints are given which govern the production of those ellipses. Justification is provided for the approach taken, and its limitations and applications are explored.

March 12, 1985

* This research was sponsored in part by the Office of Naval Research under contract N00014-80-C-0732, the National Science Foundation under grants IST-8007045 and IST-8208602, and by the Defense Advanced Research Projects Agency (DOD) ARPA Order No. 3041, monitored by the Naval Electronic Systems Command under contract N00039-82-C-0235. Special thanks go to Robert Wilensky for reading and rereading innumerable drafts of this report, and to George Lakoff for his help with some of the linguistic issues.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
12 MAR 1985		**00-00-1985 to 00-00-1985**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
The Understanding and Generation of Ellipses in a Natural Language System	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Berkeley,Department of Electrical Engineering and Computer Sciences,Berkeley,CA,94720**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Ellipsis in English can occur in many different forms. A large class of ellipses is isolated, encompassing such traditionally distinct linguistic rules as Right-node Raising, Gapping, VP-Deletion, Coordinate Reduction, and other phenomena which crosses sentence boundaries. The members of this class of ellipses involve implicit or explicit coordination. A uniform algorithm implemented in the UC (Unix Consultant) natural language system resolves the ellipses in this class, and constraints are given which govern the production of those ellipses. Justification is provided for the approach taken, and its limitations and applications are explored.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**29**	

The Understanding and Generation of Ellipses in a Natural Language System.*

Lisa F. Rau

Berkeley Artificial Intelligence Research Project
Computer Science Division
University of California, Berkeley
Berkeley, California, 94720.

1. Introduction

Consider the following examples:

1.1) Would you like to hear another verse?
I know twelve more [verses].

1.2) What is he going to do with that?
[He is going to] catch fish [with that].

1.3) I will sweep the floors today and John [will sweep the floors] tomorrow.

1.4) I asked for John's number not Mary's [number].

1.5) Mary wants white roses for the occasion.
John wants red [roses for the occasion].

1.6) Somebody came over last night -- guess when [somebody came over last night].

1.7) John loves [sunflowers] and Mary hates, sunflowers.

1.8) John loves sunflowers, and Mary [loves] John.

1.9) John was on first [base] and succeeded in stealing second [base].

---

* This research was sponsored in part by the Office of Naval Research under contract N00014-80-C-0732, the National Science Foundation under grants IST-8007045 and IST-8208602, and by the Defense Advanced Research Projects Agency (DOD) ARPA Order No. 3041, monitored by the Naval Electronic Systems Command under contract N00039-82-C-0235. Special thanks go to Robert Wilensky for reading and rereading innumerable drafts of this report, and to George Lakoff for his help with some of the linguistic issues.

(The first two examples are taken from Halliday and Hasan (1976).) These are all examples of ellipsis. Ellipsis is the omission of one or more words in sentence. Typically, this results in an incomplete grammatical construction of the sentence. Throughout this paper, words which are omitted will be bracketed. As is apparent from these examples, there are many different forms of elliptical constructions. Traditional linguistic theory distinguishes these forms according to their syntactic structure. For example, example 1.7 is classified as an instance of Right-node Raising and example 1.8, Gapping. Typically, ellipses which cross sentence boundaries, as in example 1.5, are not included in any category.

In this paper, I propose a uniform processing strategy which handles most of these forms of ellipses. The existence of this strategy suggests the elimination of the traditional distinctions between the different kinds of ellipsis. In its place, a much larger and more encompassing class of ellipsis is illustrated. This paper describes the class, and the uniform processing strategy applicable to it. The program which implements the processing strategy is illustrated with examples of its output.

2. Historical Treatment of Deletion

Winograd (1983) identifies three basic approaches to handling ellipsis. The first involves the inclusion of explicit rules for elliptical constituents. For example, the following small grammar could parse and produce ellipses in reduced sentences like:

2.1) John likes cake and Mary, cookies.

S -> S Coordinator Reduced-S
S -> NP Verb NP NP
Reduced-S -> NP NP
Reduced-S -> Verb NP NP

Unfortunately, this approach requires explicit rules for every kind of ellipsis. Also, it is incapable of handling embedded clauses. For whole phrase conjunction, however, this is an easily implementable partial solution.

The second approach, used in ATN formalisms (Woods, 1977; Boguraev, 1983) makes use of the previous history of constructions seen. It attempts to reparse from a conjunction, allowing during the reparse for elements along a path to be omitted. Among other problems, the algorithm is nondeterministic and frequently produces more than one possible parse of a given input.

The third approach involves matching constituents to recover omitted items. Many systems use one form or another of this approach. One such system is the Linguistic String Parser (Sagar, 1981; Raze, 1976). The algorithm used in the Linguistic String Parser handles only inter-sentential "conjunction reduction", with no capacity for resolving ellipses which cross sentence boundaries. It cannot handle embedded clauses. It has special rules to handle cases where matching of constituents is not exact, and where matching does not begin directly after the conjunction. This system has no concern for the production of coordinate structures.

The LIFER system (Hendrix, 1977) contains another example of this approach. In this system, there is no concern for the production of ellipses or the resolution of inter-sentential ellipsis. The mechanism is designed to facilitate repeated entry of requests from a database. Hence it is useful only when dealing with sequences of requests, as in the following examples:

2.2) Which ships have captains?
[Which] submarines [have captains]?
[Which submarines have] first-mates?

The algorithm employed in the LIFER system forces a string of words to belong to the same semantic category as some string in the previous question. It depends heavily on the presence of semantically based categories or the presence of specific case-frames to determine the "correct" meaning of the above sequences of requests. Thus a sequence like the following could not be resolved:

2.3) Which ships have captains?
Which [ships] have been assigned for duty this week?

The algorithm fails here because "been assigned for duty this week" does not belong to the same class of objects as "captains". Using semantic categories to resolve ellipsis can be helpful especially when there is more than one possible syntactic interpretation. However, it should not be the only means by which inputs must match, or the system will fail to resolve many acceptable ellipses such as example 2.3 above.

In a case like example 2.2 above, it is arguable whether the interpretations which are given are the ones a human might make. For example, the following exchange makes sense as well:

2.4) Which ships have captains?
[Which] first-mates [have captains]?

The system should have the capability to notice such potential ambiguity and perhaps notify the user.

The XCALIBUR Project (Carbonell *et. al.*, 1983) admits to not being "a general linguistic solution to the ellipsis phenomenon". It is similar in motivation to the LIFER system described above. This mechanism operates by case-frame analysis, and makes no use of syntactic clues present in the inputs. It suffers from the drawbacks that afflict the above methods.

Another system which uses some form of the matching of constituent approach is described in Grosz (Grosz, 1977). In her actual implementation, only fragmentary noun-phrases functioning as complete sentences may be resolved through syntactic and semantic matching with only the immediately preceding utterance. No inter-sentential ellipsis is resolved, or any other kind of deletion besides that occurring as the result of repeated questions.

The system also has the potential danger of producing ungrammatical ellipses. If the same rules were used to produce ellipses as understand them, the following questionable fragment might result.

2.5) Who owns all anthracite coal mines in the U.S.?
*Each natural gas pipeline?

Although extensions to the system to include other forms of deletion such as verb-phrase deletion were discussed, it was also noted that performing such modifications would greatly increase the alternatives considered for these lower level constituents during the interpretation of an utterance. This increase in the number of alternatives considered presented prohibitive problems with the speed of resolution.

The last system to be discussed is perhaps the most interesting. It has been developed by Huang (Huang, 1984), and is embedded in a definite clause

grammar (DCG) formalism based on PROLOG. Briefly, Huang's system makes use of PROLOG's pattern matching facilities by attempting to match constituents of conjuncts. When constituents in a clause of a conjunct are not present, they are "filled in" by the corresponding constituents in the other clause of the conjunct. This is triggered when the normal or unconjugated form of a sentence parse fails.

The usage of PROLOG's pattern matching facilities greatly simplifies the program needed to resolve ellipsis. Nonetheless, there are still many problems. First of all, the mechanism is only useful when dealing with inter-sentential ellipsis. Secondly, it is not clear how the mechanism could resolve ellipsis involving multiple clauses. This is related to the general problem of embedding special conditions and constraints into the uniform PROLOG pattern matcher. Lastly, the DCG approach would reconstruct some sentences as illustrated below:

2.6) She is writing to her parents today, and will again tomorrow.
*She is writing to her parents today, and will writing to her parents again tomorrow.

The algorithm's failure to process such sentences correctly is due to the inflexibility with which constituents must match. Constituents are matched exactly, without consideration of such things as proper tense and agreement.

The above survey suggests that ellipsis is not a solved problem. The rest of this paper is an attempt to provide a fairly simple and straightforward yet general solution to the problem of understanding ellipses.

3. Classification of Ellipsis

The class of ellipsis which the algorithm given in the next section can resolve is a class composed of explicit and implicit coordinate structures. Explicit coordinate structures have been hypothesized by some computational linguists *(cf.* Huang, 1984; Raze, 1976; Dik, 1968) to encompass traditional linguistic transformations such as Right-node Raising, Gapping, and VP-deletion. Explicit coordinate structures differ from implicit coordinate structures in that explicit coordinate structures contain an explicit coordinator in the input. Below is an example of an explicit and an implicit coordinate structure.

Explicit Coordinate Structure:

I like rice and Mary, beans.

Implicit Coordinate Structure:

How do you get to school from here?
How does John?

This class of coordinated structures is the focus of the remainder of the paper. There are other kinds of phenomena which might be considered elliptical, but which are not included in the class of coordinated structures. This exclusion is motivated by the difference in *processing strategy* which must be employed to understand the input. I isolate two additional classes of ellipses which cannot be resolved with the processing strategy used to resolve the class of coordinated structures. The first is the class of ellipses which arise as answers to questions, and the second is the class of ellipsis which involves the omission of words not literally (i.e., lexically) present anywhere in the discourse.

The mechanisms used to understand answers to questions are different from those used to understand coordinate structures. In the case of coordinate structures, as we shall see from the next section, a match-and-insert process is used,

augmented with special linguistic heuristics to construct a new input capable of being analyzed by the natural language analyzer. In the case of question-answer pairs, the process of understanding the answer to the question involves determination of empty functional slots in the sentence which corresponds to the question, and inserting the information obtained in the answer.

The proposed differences in processing strategy to resolve coordinate structures and question-answer pairs seems clear. This contrasts with the somewhat vaguer distinction between coordinate structures, where one of the coordinated clauses contains material which is used in understanding another coordinated clause, and the case of *metonymy*, where the material used to understand the ellipses is not necessarily literally present in the discourse. The following examples are examples of this phenomena.

3.1) Mary won the bronze in Sarajevo.

3.2) Tom was on first, and succeeded in stealing second.

3.3) John has a Picasso hanging in his bedroom.

3.4) Mary went to Macy's and bought a Dior.

The understanding mechanism which is to resolve these examples must have a method of determining *from the surrounding context* what material must be present to understand the input. This would be a different kind of algorithm than an algorithm which looks to specific lexical items in the discourse as candidates for omitted material.

One might suppose that all that would be needed to resolve these cases of metonymy is for a general algorithm to look in a record of the current context as opposed to a record of the current utterances for absent material. But this will not always yield a correct referent.

For example, consider sentence 3.3. Given that the system knows that "Picasso" is the name of a painter, it might incorrectly assume that this painter was hung on the wall. General knowledge of the situation and knowledge of preferred ways of referring to some objects is needed to correctly interpret this sentence as referring to a painting done by Picasso. Simple syntactic matching would provide no such information. In sum, to determine what are appropriate candidates for the missing material, one must be able to distinguish relevant from irrelevant items.

The distinction between ellipses and metonymy is sometimes blurred. However, if a language understander can understand sentences like 3.3 and 3.4 through knowledge of the domain or common ways of referencing items like paintings, the resolution mechanism for ellipses will not be needed. The same holds true for cases of conjoined subjects, or other conjoined parts-of-speech. For example, given:

3.5) John and Mary are engaged.

3.6) Mary greeted or dismissed the callers.

In these two cases, the problem of understanding the combinatory or segregatory coordination is left as a problem for the natural language understander.

To summarize the classification proposed in this section, it has been suggested that the class composed of explicit and implicit coordinate structures is fundamentally different with respect to processing strategies from other elliptical constructions. Some of these other constructions are ellipsis which arises as an answer in a question-answer pair, and certain kinds of metonymy, where items are referred to which are not lexically present in the discourse.

4. Algorithm to Resolve Coordinated Deletions

In Section 2, we saw how previously proposed solutions to the resolution of ellipsis had certain drawbacks. The main problems can be summarized as follows:

1) No generality or breadth in the solutions - some systems only understand sequences of questions, others only understand inter-sentential gapping.

2) Results which are ambiguous - one input may lead to many possible interpretations.

3) Incorrect interpretations of inputs resulting from inflexible matching processes which ignore tense and agreement.

4) No concern for the detection of ungrammatical ellipsis.

5) No concern for the production of ellipses in general.

In this section, I give an English language description of heuristics to be applied to interpret and to produce coordinated ellipses. The algorithm to be described addresses all of the above problems. It is a general solution which can resolve sequences of questions, inter-sentential ellipsis, and other forms which ellipses may occur in. Separate heuristics may be applied to govern the production of ellipsis. These heuristics make detection of ungrammatical inputs possible. This gives the analyzer the option of flagging the user when an input is potentially inherently ambiguous or ungrammatical. In addition, these heuristics help to ensure that no ungrammatical ellipses will be produced.

The algorithm is called into play when, in understanding, some input sentence cannot be analyzed with ordinary application of understanding processes. In production, it would be triggered by the presence of a concept to be expressed, parts of which are identical to each other.

The algorithm consists of four main phases. The first is an information-collecting phase. Information about the type of coordination and the boundaries

of the clauses is recorded for use in breaking up the sentence or sentences into its constituent phrases.

In the second phase, constraints on the acceptability of the ellipsis are checked. These constraints are the exceptions to the general rule of ellipsis, and are applied to supress the production of ungrammtical or unnatural ellipses. These constraints are discussed in a later section of this paper.

The third phase consists of a "match-and-insert" process used to resolve an ellipsis, or a deletion process used to produce an ellipsis. This is followed by the last phase, in which the sentence is reanalyzed or the ellipsis is produced.

ALGORITHM

1) **Determine if the utterance was in response to a question.**

   If so, we do not use this heuristic to resolve the ellipsis.

2) **Check for the presence of coordinators, subordinators, clause containers, other special words and punctuation.**

   Coordinators: and, but, or, nor, for.

   Correlative conjunctions: either-or, neither-nor, both-and, not-only-but-also.

   Subordinators: after, although, as if, as long as, as soon as, because, before, if, in order that, since, so that, than, though, unless, until, when, whenever, where, wherever, while

   Inversion Triggers: neither, nor, so

   Boundaries: sentence boundaries, intonation clues if present.

   Punctuation: commas, semi-commas, periods, question marks, exclamation marks.

   The algorithm uses this information in step 4 to determine where the coordinated "thoughts" occur.

   As a result of this step, the algorithm has more information to be used later in determining where the boundaries of the conjoined thoughts or phrases are, and the applicability and use of the heuristic.

3) **Check for constraints.**

These constraints may be found in Section 6 of this paper.

If we are producing an ellipsis, specific constraints are applied, using the information obtained above.

If the sentence cannot comply with the constraints, we do not produce the ellipsis.

In understanding, inputs which violate the constraints are ungrammatical. If the match-and-insert process can resolve the ellipsis in spite of the ungrammaticality, then we have the option of resolving the input, or rejecting it with a flag to the user that the input was unacceptable.

4) **Divide sentence into appropriate clauses.**

Given the information obtained in step 1, we can divide the input into clauses. For example, given:

4.1) John went to the store and the laundromat.

We subdivide the sentence into the following clauses:

4.2) John went to the store
the laundromat

Similarly, we have:

4.3) What is the difference between Japanese eggplant and Chinese?

4.4) What is the difference between Japanese eggplant
Chinese

4.5) Did John go to New York?
To Los Angeles?

Note that in the last example, the sentences are already divided into the correct components. In all the examples I have come across, we can appropriately divide the input into the different subclauses or parts by considering the presence or absence of the information collected in steps 2 and 3 of this algorithm.

5) **If understanding, begin matching.**
If producing, omit.

Understanding:

The matching process begins at the first words of each clause. Words which are identical lexical items are matched together and an insertion process begins. If there are no words which match exactly, a match is performed to find the first words which have the same part-of-speech, and then the

insertion process begins.

If there is more than one possible match, semantic category information is used to pick the best match.

In the insertion process, missing word are inserted before the matched word or words, up to the clause container or the beginning of the sentence.

Continue matching after this word.

Allow subject-auxiliaries to match with auxiliaries-subject if there was an inversion trigger, to resolve examples like:

4.6) Mary will go, and so will John [go].

Complete phrases or constituent as appropriate by insertion.
Stop after phrase or constituent has been completed.

If this last insertion took place at the end of the first clause, then no words after the completion are inserted from the end of the sentence.

This allows for the correct resolution of sentences like:

4.7) Brian wrote to his parents and [Brian] will be writing again today.

Pronouns are considered to match with noun phrases, when they occur in the identical positions in the clauses. For example, given:

4.8) John likes rice however she doesn't [like] beans.

The noun phrase "John" would match part-of-speech with the pronoun "she".

If a verb without subject is introduced, check agreement and correct if necessary. This heuristic looks not only at the subject but any auxiliaries that might be retained. For example, the following sentences require changes with agreement.

4.9) What are the prices of your two largest computers?
Speed?
[What is the] speed [of your two largest computers]?

4.10) She is writing to her parents today and will again tomorrow.
She is writing to her parents today and will [write to her parents] again tomorrow.

Negatives are considered "attached" to the verb and form one indivisible unit for matching purposes.

Auxiliaries are only inserted from the first clause to the second.

If the coordinator is a negative coordinator such as *neither* or *nor*, double negatives present in the first clause like "not" and "no" are not inserted in the second clause.

Producing:

The production of ellipsis is governed by constraints on where identical material occurs in the complete sentence to be produced. These constraints, although fairly complicated, give guidelines for the determination of whether the identical material is more naturally omitted from, for example, the first clause or the second clause of the sentence to be produced. These constraints have not yet been added to the UC system. Detailed descriptions of the constraints used to produce ellipsis may be found in (Ross, 1970) and (Van Oirsouw, 1984).

6) **Reanalyze sentence or produce sentence.**

Understanding:

Reanalyzation of the sentence is performed anew. Although some of the information obtained from the first analyzation could be useful, some form of new analysis must be performed to obtain a complete and correct conceptual representation of the input.

In production:

The ellipsis is produced to the user.

It is important to note that the understanding algorithm and the production algorithm operate on different inputs, and give different outputs. This makes for an inevitable difference in processing strategy. In the production of ellipses, we assume a full sentence containing identical parts, and use clues to determine what to omit from the actual production of the utterance. In understanding, the process of analyzing an ellipsis involves determining repeated structure and from that, filling in what is necessary to make the coordinated structures match. However, constraints which govern the acceptability of ellipsis as discussed in Section 6 are accessed by both the production algorithm and the understanding algorithm. The constraints are used by the production algorithm to inhibit production of unacceptable ellipses, and by the understanding component to detect ungrammatical inputs.

Example

Let us examine a trace of the understanding component of the algorithm on two examples:

4.11) The difference between the tax on earned and on unearned income is enormous.

The algorithm notes the presence of the keywords "and" and "between" and breaks the sentence into three parts according to where explicit clause containers as listed above occur:

The difference between
the tax on earned
on unearned income is enormous.

Now the identical lexical items are identified. The word "on" appears first in both clauses. The insertion process then inserts whatever words are missing from the second clause, in this case, the words "the tax". Now we try to match identical words, but no more exist. The algorithm then tries to match identical part-of-speech. The words "earned" and "unearned" match by this criterion. Insertion is then performed again to complete the phrase in the first clause in the same way as in the second clause. In order to complete the noun-phrase, the algorithm inserts the word "income" to the first clause. Now that the phrase is complete, the insertion terminates.

The next example also illustrates some of the power of the algorithm.

4.12) They no doubt can and should but probably won't pay the full fee.

In this example, we see that the heuristic must be applied to three clauses. The algorithm begins the matching always from the last clause to the first, as, when there are multiple clauses, the realized constituents always appear in the last clause. Thus initially the algorithm breaks the sentence up into its three

constituent clauses:

They no doubt can
should
probably wont pay the full fee

After breaking the sentence up into its constituent clauses, the algorithm proceeds to match the first clause to the last clause as follows:

probably wont pay the full fee
They no doubt can

which yields:

They no doubt can pay the full fee.

Note that the modifier "probably" matches "no doubt". Given the above matched constituent, we match:

They no doubt can pay the full fee

should

will yield:

They should pay the full fee

to get the full sentence. Remembering that we consider all adverbials to be "attached" to the auxiliary and that the resulting unit matches auxiliaries, "no doubt" will be attached to "can", and the resulting adverbial- auxiliary pair "no doubt can" will match the auxiliary "should".

5. Implementation

The understanding component of the ellipsis algorithm is coded in Franz LISP. It runs as a separate module called by the PHRAN analyzer (Wilensky and Arens, 1980) as part of the UC (Unix Consultant) system (Wilensky, Arens and Chin, 1984) in the event that the PHRAN cannot successfully parse the input. The generator PHRED (Jacobs, 1983) does not yet access the constraints on the production of ellipses in forming its output. The following is a small sample of ellipsis resolution by the algorithm.

To a UC '#' prompt, please type in your questions about Unix in English. To leave, just type a ''D' or '(exit)'.

Hello.
How can I help you?

How do I delete a file?
Typing 'rm filename' will remove the file with name filename from your current directory.

A directory?
Typing 'rm -r directoryname' will recursively delete the directory with name directoryname and all the file and directories contained in the directory.

What is the best way to create a file ?
The best way to create a new file is to use the editor, i.e.- 'ex filename'.

A directory?
Typing 'mkdir directoryname' will create a directory with name directoryname in your current directory.

How can I sort a file?
Type 'sort file1' to sort the file with name file1.

Compare two files?
To compare two files, type 'diff file1 file2'.

What is a directory?
A directory in Unix is a special file which contains other files and directories. This provides a hierarchical format for naming files by giving the path of directories in which a file is contained. Such a name where directories are separated by '/' is called a path name (q.v.).

An operating system?
An operating system is a large program(s) that serves as an interface between the user and the machine and which provides a collection of helpful utilities such as a

file system(q.v.).

The implementation takes only a tiny fraction of the total code and total time used for parsing. The module is completely independent of the parser, and needs only the partial syntactic analysis of the input and a copy of the previous input as typed by the user and its parse to operate.

6. Constraints

In this section, some commonly accepted constraints on ellipsis are illustrated. These are the constraints referred to in Section 4, part 4 of this paper. The production component will reject an elliptical construction that violates one of these constraints. The understanding component of the system has the option of notifying the user that the input was ungrammatical, or silently reconstructing the questionable interpretation. An exception to these constraints would cause the production component to generate a potentially unusual sounding sentence, or would cause the understanding component to flag an acceptable sentence as questionable. The examples are from Halliday and Hasan (1976), and from Lanacker (1963).

1) Do not produce more than two noun phrases in a row.

   6.1) *I bought Sally roses and Jack [bought] Jane lilies.

2) Only elide subject and auxiliaries, or auxiliaries only, from clauses subsequent to the first clause.

   6.2) Peter must have broken in and [Peter must have] stolen the papers.
   *[Peter must have] broken in and Peter must have stolen the papers.

   6.3) Peter must have broken in and John [must have] stolen the papers.
   *Peter [must have] broken in and John must have stolen the papers.

3) Do not elide subject and/or auxiliaries or verb when there are two coordinated clauses, the second of which contains a subordinate clause.

   6.4) *Peter must have broken in and I'm sure that [Peter must have] stolen the papers.

   6.5) *John must clean the shed and it seems that Peter [must] read his book.

   6.6) *Paul likes Mary and I know that Peter [likes] Joan.

4) Elide only auxiliaries only if the subject is coreferential in the two clauses. It is most natural in this case to delete all the auxiliaries.

6.7) John must clean the shed and Peter [must] read his book.

6.8) *Peter must clean the shed and Peter [must] read his book.

6.9) *Peter must clean the shed and he [must] read his book.

6.10) *Peter may be cleaning the shed and Peter may [be] reading his book.

7. Summary and Conclusions

I have proposed that ellipses which belong to the class of coordinated structures may be mostly resolved with a uniform processing strategy. This processing strategy does not apply to ellipses which arise as answers to questions and which are members of a small class of metonymic utterances.

The proposed algorithm resolves the ellipses belonging to the class of coordinated structures. The algorithm uses surface syntactic clues, word order information and semantic category information to perform the resolution. Effective constraints are included for production of ellipses, and for the detection of ungrammatical inputs, if desired. A discussion of one of the problems with this approach was also given.

Not only does the strategy of match-and-insert of constituent structures work, it succeeds in accomplishing two important objectives. First, it condenses previously considered separate linguistic transformations into one phenomenon. Gapping, Coordinate Reduction, Verb-phrase Deletion, and Right-node Raising, for example, are all treated in a uniform manner. This uniformity in processing suggests that previously considered separate rules in the linguistic sense might not be separate phenomenon at all. The approach is also useful in resolving ellipses which occur across sentence boundaries.

Secondly, this approach proposes a new theory of the formation and understanding of utterances. Implicit in this approach is the belief that in understanding, *patterns* of syntactic information, along with semantic information, are utilized to "fill in" perceived *gaps* in the grammatical structure of utterances. In production, it is proposed that an utterance must be formed completely before the production of ellipses may take place. After formation, the omission of words occurs as a result of perceived *redundancy* in the surface structure. Thus a heuristic which says "don't say more than you have to" could be thought to be

at work.

Also implicit in this approach is the belief that ellipses which contain matching structures necessitate a different kind of processing strategy from that which the normal language understander does. This process specifically utilizes the presence of the matching structures in the ellipsis. The presence of such a separate strategy suggests that other linguistic phenomena which are difficult to handle by an extension of the "core grammar" in the system might also be better or more completely handled by a processing component specifically tailored to that phenomenon. Some candidates for such an approach might be the understanding of answers to questions, and the understanding of various kinds of ill-formed input.

8. Appendix

In this appendix, I will give examples of many different kinds of ellipses which the algorithm given can resolve. Most of the examples and the classification scheme are taken from Halliday and Hasan (Halliday and Hasan, 1976). These examples are not all parsed by PHRAN due to limitations of the database. However simulation via attachment of the appropriate syntactic and semantic category information has allowed the algorithm to be actually tested on most of these examples.

ELLIPSIS OF LEXICAL VERB

8.1) She has written to her parents and he may [write] to his sister.

8.3) Alice was happy and Susan [was] miserable.

8.4) I work on a farm and my brother [works] in a factory.

8.5) She is writing to her parents and [she] will be [writing] to her brother.

ELLIPSIS OF VERB INCLUDING AUXILIARY

8.6) Paul is flying to N.Y. tomorrow and [Paul is flying] to L.A. next week.

ELLIPSIS OF VERB AND SUBJECT COMPLEMENT

8.7) John was the winner in 1970 and Bob [was the winner] in 1971.

8.8) It's cold in December in New England but [it's cold] in July in New Zealand.

ELLIPSIS OF VERB AND OBJECT

8.9) Mary will cook the dinner today and Joan [will cook the dinner] tomorrow.

ELLIPSIS OF AUXILLARIES

Present and modal

8.10) John understands the situation and surely Peter should [understand the situation].

Past and modal

8.11) Bob entered the competition and Paul may [enter the competition].

Perfect and modal

8.12) John hasn't met my brother yet but [he] will [meet my brother] soon.

Progressive and modal

8.13) Peter is complaining about the noise but John won't [complain about the noise].

Progressive and perfect

8.14) John is questioning our motives and Bill has [questioned] our results.

Past and perfect

8.15) Peter saw your parents last week, but [he] hasn't [seen your parents] since.

ELLIPSIS OF ADVERBIAL

8.16) Tom was at Oxford, but his brother wasn't [at Oxford].

8.17) Brian wrote to his parents and [Brian] will be writing [to his parents] again today.

8.18) He spoke for the first [motion] and against the second motion.

ELLIPSIS OF HEAD OF NOUN PHRASE

8.19) We wanted fried fish, but they gave us boiled [fish].

8.20) She will drive to [London], but [she will] fly back from London.

8.21) He was a friend to [the party leader], and [he was] a strong supporter of, the party leader.

ELLIPSIS OF COMPLEMENT OF PREPOSITIONAL PHRASE

8.22) John crawled under [the fence], but Bill climbed over, the fence.

8.23) He walked up [the hill], and [he] ran down, the hill.

ELLIPSIS IN REPEATED QUESTIONS

8.24) How are you going to school?
How is John?

8.25) What kind of ice cream do you like?
What kind of candy?

8.26) Where do you put your clothes?
And your shoes?

ELLIPSIS IN RELATED UTTERANCES

8.27) Mary thinks vanilla is the best kind of ice cream.
John thinks chocolate.

8.28) I ordered white wine.
Not red.

8.29) Put the ball on the table.
Now the cube.

8.30) This is a big elephant with floppy ears.
That isn't.

9. References

Akmajian, Adrian and Frank Heny. *An Introduction to the Principles of Transformational Syntax.* MIT Press, Cambridge, Mass, 1975.

Boguraev, B. K. Recognizing conjunctions without the ATN framework. *Automatic Natural Language Parsing.* (Editored by: Sparck-Jones, K. and Wilks, Y.). Ellis Horwood, 1983.

Carbonell, J, W. Boggs, M. Mauldin, and P. Anick. XCALIBUR Progress Report #1, Overview of the Natural Language Interface. Technical Report, Carnegie-Mellon University, Computer Science Department, 1983.

Chomsky, Noam. *Aspects of a theory of syntax.* MIT Press, Cambridge, Mass, 1965.

Dik, Simon. *Coordination and its implication for the theory of general linguistics.* North-Holland Publishing Company, Amsterdam, 1968.

Dougherty, R. A Grammar of Coordinate Conjoined Structures. *Language,* Volume I. pp. 850-898, Vol. 46, 1970.

Gass, William. "And". *Harper's,* February, 1980.

Green, Bertram, A. K. Wolf, C. Chomsky, and K. Laughery. BASEBALL: An automatic question-answerer. In *Computers and Thought* pp. 207-216, McGraw-Hill, New York, 1963. (Editors: Edward A. Feigenbaum and Julian Feldman)

Grinder, John and Paul Postal. Missing Antecedents. *Linguistic Inquiry,* Vol. II.3, 1971.

Grosz, Barbara. The Representation and Use of Focus in Dialogue Understanding. Technical Note #151, SRI International, 1977.

Hankamer, Jorge. Unacceptable Ambiguity. *Linguistic Inquiry,* Vol. 4 pp. 17-68, 1973.

Halliday M.A.K., and Ruqaiya Hasan. *Cohesion in English.* Longman, New York, 1976.

Hendrix, Gary. The LIFER Manual: A guide to Building Practical Natural Language Interfaces. Technical Report, Note #138, SRI International, 1977.

Huang, Xiuming. Dealing with conjunctions in a Machine Translation Environment. In *Proceedings of the 10th International Conference on Computational Linguistics,* Stanford University, Palo Alto, California, July, 1984.

Jackendoff, Ray. Gapping and Related Rules. *Linguistic Inquiry,* Vol. 2, pp. 21-35, 1971.

Jacobs, Paul. Generation in a Natural Language Interface. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence.* Karlsruhe, F.R.G. 1983.

Jameson, Anthony and Wahlster, Wolfgang. User Modelling in Anaphora Generation: Ellipsis and Definite Descriptions. In *Proceedings of the First European Conference on Artificial Intelligence* Orsay, 1982

Jameson, Anthony. Documentation for three HAM-ANS Components: Ellipsis, NORMALIZE and NORMALIZE-1. MEMO ANS-4, November, 1981.

Kuno, Susumo. Gapping: A Functional Analysis. *Linguistic Inquiry,* Vol. 7, pp. 300-318, 1976.

Lakoff, George. Symmetric Predicates. *Modern Studies in English.* Readings in Transformational Grammar, 1966.

Langendoen, D. Acceptable Conclusions from Unacceptable Ambiguity. In *Proceedings from Testing Linguistic Hypotheses,* University of Wisconsin, Milwaukee, Wisconsin, May, 1974.

Lindsay, Robert. In defense of ad hoc systems. In *Computer Models of Thought and Language,* pp. 372-395, Freeman, San Francisco, 1973. (Edited by: Roger C. Schank, Kenneth M. Colby and Lindsay).

Postal, Paul. *On Raising: One rule of English grammar and its theoretical implications.* MIT Press, Cambridge, Mass, 1974.

Raze, Carol. A Computational Treatment of Coordinate Conjunction. *American Journal of Computational Linguistics,* Microfiche #52, 1976.

Ross, Robert. Gapping and the Order of Constituents. *Progress in Linguistics,* 1970, Edited by Bierwirsch and Heidolph, The Hague: Mouton.

Ross, Robert. Guess Who? Fifth Regional Meeting, Chicago Linguistics Society, Chicago, Illinois, 1969 (Edited by: R. Binnick, et. al.).

Sagar, Naomi. *Natural Language Information Processing.* Addison-Wesley, Reading, Mass, 1981.

Tai, J. *Coordinate Reduction.* Indiana University Linguistics Club, 1969.

Van Oirsouw, R. Untitled manuscript. University of Utrecht, Holland, 1984.

Wilensky R., and Arens, Y. *PHRAN - A Knowledge-based Approach to Natural Language Analysis.* University of California at Berkeley, Electronics Research Laboratory Memorandum #UCB/ERL M80/34, 1980.

Wilensky, R., Arens, Y. and Chin, D. Talking to UNIX in English: An Overview of UC. *Communications of the ACM.* Volume 27, Number 6, pp. 574-592, 1984.

Winograd, Terry. *Language as a Cognitive Process, Vol. 1 - Syntax.* Addison-Wesley, Reading, Mass, 1983.

Woods, William. Semantics and Quantification in Natural Language Question Answering. Technical Report #3687, Bolt, Beranak and Newman, Cambridge, Mass, November, 1977.